

TUGAS KECIL 2 IF2211 STRATEGI ALGORITMA
PENYUSUNAN RENCANA KULIAH DENGAN *TOPOLOGICAL SORT*
(PENERAPAN *DECREASE AND CONQUER*)

Oleh

La Ode Rajuh Emoko - 13519170



PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

2020/2021

1. Deskripsi Aplikasi

Aplikasi ini dapat Menyusun rencana pengambilan kuliah dengan memanfaatkan algoritma Decrease and Conquer. Penyusunan Rencana Kuliah diimplementasikan dengan menggunakan pendekatan Topological Sorting.

Aplikasi akan menerima daftar mata kuliah beserta prasyarat yang harus diambil seorang mahasiswa sebelum mengambil mata kuliah tersebut. Daftar mata kuliah tersebut dituliskan dalam suatu file teks dengan format:

```
<kode_kuliah_1>,<kode kuliah prasyarat - 1>, <kode kuliah prasyarat - 2>, <kode kuliah prasyarat - 3>.  
<kode_kuliah_2>,<kode kuliah prasyarat - 1>, <kode kuliah prasyarat - 2>.  
<kode_kuliah_3>,<kode kuliah prasyarat - 1>, <kode kuliah prasyarat - 2>, <kode kuliah prasyarat - 3>,<kode kuliah prasyarat - 4>.  
<kode_kuliah_4>.
```

Sebuah kode kuliah mungkin memiliki nol atau lebih prasyarat kuliah. Kode kuliah bisa diambil pada suatu semester jika semua prasyaratnya sudah pernah diambil di semester sebelumnya (tidak harus 1 semester sebelumnya). Asumsi semua kuliah bisa diambil di sembarang semester, baik semester ganjil maupun semester genap, dan tidak ada yang *looping* (contoh: mata kuliah A butuh prasyarat B, mata kuliah B butuh prasyarat C, mata kuliah C butuh prasyarat A).

Dari file teks yang telah diterima, ditentukan kuliah apa saja yang bisa diambil di semester 1, semester 2, dan seterusnya. Sebuah kuliah tidak mungkin diambil pada semester yang sama dengan prerequisitenya. Untuk menyederhanakan persoalan, tidak ada Batasan banyaknya kuliah yang bisa diambil pada satu semester.

Aplikasi akan memberikan keluaran berbentuk dengan format

Semester 1: mata kuliah 1, mata kuliah 2, mata kuliah 3, dst.

Semester 2: mata kuliah 1, mata kuliah 2, dst.

Dst...

2. Algoritma Topological Sort dan Pendekatannya Terhadap Algoritma Decrease and Conquer

Langkah pertama adalah mengambil data mata kuliah dari file txt dan membentuk directed acyclic graph. Pembentukan graph yang digunakan pada aplikasi ini adalah dengan mencatat banyaknya mata kuliah prasyarat dari setiap matkul, serta mencatat semua mata kuliah yang ditunjuk oleh masing-masing mata kuliah. Dengan kata lain, menghitung jumlah sisi yang masuk pada setiap simpul serta mencatat simpul-simpul yang ditunjuk oleh masing-masing simpul. Hal ini dapat dilakukan menggunakan dictionary pada python. Untuk lebih jelasnya perhatikan masukan dan keluaran dibawah ini:

```
C1, C3.  
C2, C1, C4.  
C3.  
C4, C1, C3.  
C5, C2, C4.
```

Figure 2 contoh input file

```
banyak matkul prasyarat dari matkul C1 : 1  
banyak matkul prasyarat dari matkul C2 : 2  
banyak matkul prasyarat dari matkul C3 : 0  
banyak matkul prasyarat dari matkul C4 : 2  
banyak matkul prasyarat dari matkul C5 : 2  
  
matkul C1 merupakan prasyarat dari: C2 , C4 ,  
matkul C2 merupakan prasyarat dari: C5 ,  
matkul C3 merupakan prasyarat dari: C1 , C4 ,  
matkul C4 merupakan prasyarat dari: C2 , C5 ,  
matkul C5 merupakan prasyarat dari:
```

Figure 1 Graph hasil dari contoh input pada Figure 2

Selanjutnya untuk setiap semester, diambil mata kuliah dengan jumlah prasyarat yang belum diambil sebanyak 0. Artinya diambil semua mata kuliah yang semua prasyaratnya sudah terpenuhi. Dalam pendekatan terhadap algoritma *Decrease and Conquer*, tahap ini masuk kepada tahap *conquer*.

Kemudian setelah mata kuliah diambil, semua sisi yang berasal dari mata kuliah tersebut harus dihilangkan. Ini dilakukan dengan cara mengurangi derajat simpul masuk dari semua mata kuliah yang ditunjuk oleh mata kuliah yang diambil sebanyak 1. Langkah ini adalah bagian dari tahap *decrease* pada pendekatan algoritma *Decrease and Conquer* karena kita mengurangi pengecekan dengan membuang case sisi yang tidak akan kita gunakan lagi.

Sebagai ilustrasi, saat kondisi pada figure 1 diambil C3, kondisi sekarang menjadi:

```
===== RESTART: D:\Kuliah\Semester 4\Strategi Algoritma
banyak sisa matkul prasyarat dari matkul C1 : 0
banyak sisa matkul prasyarat dari matkul C2 : 2
banyak sisa matkul prasyarat dari matkul C3 : 0
banyak sisa matkul prasyarat dari matkul C4 : 1
banyak sisa matkul prasyarat dari matkul C5 : 2
>>> |
```

Figure 3 ilustrasi pemotongan sisi graph setelah matkul C1 diambil

Dapat dilihat bahwa sisa prasyarat dari matkul C1 dan C4 berkurang satu karena mata kuliah C3 merupakan prasyarat dari mata kuliah C1 dan C4, sehingga setelah C3 diambil, sisi dari C3 ke C1 ataupun ke C4 dihapus.

Setelah menghapus sisi yang terhubung dengan C3, barulah C3nya dihapus. Langkah ini juga merupakan bagian dari tahap *decrease* karena kita mengurangi pencarian dengan menghapus mata kuliah yang sudah diambil.

Selanjutnya tinggal mengulangi Langkah dari mengambil mata kuliah yang prasyaratnya sudah terpenuhi sampai menghapus mata kuliah yang sudah diambil untuk semester semester selanjutnya hingga data banyaknya kuliah prasyarat sudah kosong karena semuanya sudah dihapus(diambil)

Program utama dari aplikasi ini ada pada gambar dibawah ini.

```
1 ##La Ode Rajuh Emoko
2 #13519170
3 #K4
4 #main.py
5
6 import bacafile13519170 as BF
7 import KurangiPrasyaratDanHapusEdge13519170 as KP
8
9 f = open("test.txt", "r")
10 teks = f.readlines()
11 banyakPrasyarat = {}
12 akuPrasyaratDari = {}
13
14 matkulDiAmbil = [[] for i in range(8)]
15
16 BF.bacafile(teks, banyakPrasyarat, akuPrasyaratDari)
17
18 for i in range(8): #untuk masing masing semester
19     if(len(banyakPrasyarat) == 0): break #jika semua matkul sudah diambil, maka tidak ada lagi matkul yang bisa diambil, iterasi berhenti
20     else: #masih ada matkul yang belum diambil
21         #mengambil matkul(key) yang banyak prasyarat yang belum diambilnya = 0 (semua prasyarat telah diambil)
22         matkulDiAmbil[i] = [key for key in banyakPrasyarat if banyakPrasyarat[key] == 0]
23
24         for key in matkulDiAmbil[i]: #untuk matkul yang sudah diambil di semester i+1,
25             #Kurangi jumlah prasyarat dari matkul matkul yang lainnya yang mempunyai prasyarat matkul key
26             KP.KurangiPrasyaratDanHapusEdge(banyakPrasyarat, akuPrasyaratDari, key)
27
28             del banyakPrasyarat[key] #hapus matkul yang sudah diambil (sudah dimasukkan di semester ke i+1)
29
30 for i in range(len(matkulDiAmbil)):
31     print("Semester ", i+1, ":", matkulDiAmbil[i][0], end='')
32     for j in range(1, len(matkulDiAmbil[i])-1):
33         print(", ", matkulDiAmbil[i][j], end='')
34     print(", ", matkulDiAmbil[i][len(matkulDiAmbil[i]) - 1])
35 ~
```

Figure 4 program utama

3. Hasil Input dan Output

a. Testcase 1

```
C: > Users > rajuh > Documents > GitHub > Tucil-2-Stima > src > test.txt
1 C1, C3.
2 C2, C1, C4.
3 C3.
4 C4, C1, C3.
5 C5, C2, C4.

IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
===== RESTART: C:\Users\rajuh\Documents\GitHub\Tucil-
Semester 1 : C3
Semester 2 : C1
Semester 3 : C4
Semester 4 : C2
Semester 5 : C5
>>> |
```

Figure 5 testcase 1

b. Testcase 2

```
> Users > rajuh > Documents > GitHub > Tucil-2-Stima > src > test.txt
1 Kriptografi, Matdis.
2 Kalkulus.
3 TBFO, Matdis.
4 Fisika.
5 Stima, Matdis, Kalkulus.
6 Matdis, Kalkulus.

IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
===== RESTART: C:\Users\rajuh\Documents\GitHub\Tucil-
Semester 1 : Kalkulus, Fisika
Semester 2 : Matdis
Semester 3 : Kriptografi, TBFO, Stima
>>> |
```

Figure 6 testcase 2

c. Testcase 3

```
> Users > rajuh > Documents > GitHub > Tucil-2-Stima > src > test.txt
1 MA1201, MA1101.
2 FI1201, FI1101.
3 IF1210, KU1102.
4 KU1202, KU1102.
5 KI1002, KU1011.
6 EL1200, FI1101.
7 KU1102.
8 MA1101.
9 FI1101.
10 KU1011.

IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
===== RESTART: C:\Users\rajuh\Documents\GitHub\Tucil-2-Stima\src\ma
Semester 1 : KU1102, MA1101, FI1101, KU1011
Semester 2 : MA1201, FI1201, IF1210, KU1202, KI1002, EL1200
>>> |
```

d. Testcase 4

```
C: > Users > rajuh > Documents > GitHub > Tucil-2-Stima > src > test.txt
1   Flask, Python, Pip.
2   Pip, Python.
3   Python, C.
4   C.

IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
itHub\Tucil-2-Stima\src\main.py =====
Semester 1 : C
Semester 2 : Python
Semester 3 : Pip
Semester 4 : Flask
~ ~ ~
Figure 7 testcase 4
```

e. Testcase 5

```
C: > Users > rajuh > Documents > GitHub > Tucil-2-Stima > src
1   B1.
2   C2, B1.
3   A3, C2.
4   B4.
5   C5.
6   A6, A3, B4, C5.
7   B7, C5, C2.
8   C8, C5, A3, B1.
9   A9, B7, C8, C2, C5, A6, B1.
10  B10, C2, B7, A9, C8, B1, A6, C5.
11  C11, C5, B7.
12  A12, C11, B4, C8, B10.
13  B13, C2, A3, B7, A6, C5, C11, A12.
14  C14, C5, B7.
15  A15, B4, A9.

IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
===== RESTART: C:\Users\rajuh\Documents\Gith
Semester 1 : B1, B4, C5
Semester 2 : C2
Semester 3 : A3, B7
Semester 4 : A6, C8, C11, C14
Semester 5 : A9
Semester 6 : B10, A15
Semester 7 : A12
Semester 8 : B13
~ ~ ~
Figure 8 testcase 5
```

f. Testcase 6

```
C: > Users > rajuh > Documents > GitHub > Tucil-2-Stima > src > test.txt
1 B1.
2 C2.
3 A3.
4 B4, A3, B1.
5 C5.
6 A6.
7 B7.
8 C8, A3, B7, A6.
```

```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
>>>
===== RESTART: C:\Users\rajuh\Documents\GitHub\Tucil-
Semester 1 : B1, C2, A3, C5, A6, B7
Semester 2 : B4, C8
>>> |
```

Figure 9 testcase 6

g. Testcase 7

```
C: > Users > rajuh > Documents > GitHub > Tucil-2-Stima > src > test.txt
1 B1.
2 C2.
3 A3.
4 B4, B1, A3.
5 C5.
6 A6, C5, B4.
```

```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
===== RESTART: C:\Users\rajuh\Documents\GitHub\Tucil-
Semester 1 : B1, C2, A3, C5
Semester 2 : B4
Semester 3 : A6
>>> |
```

Figure 10 testcase 7

h. Testcase 8

```
C: > Users > rajuh > Documents > GitHub > Tucil-2-Stima > src > test.txt
1 B1.
2 C2, B1.
3 A3, B1, C2.
4 B4, C2, A3.
5 C5, B4.
6 A6, C2, A3, B1.
7 B7.
8 C8, A6.
9 A9, B4, C2, B1, C8, C5.
10 B10, C2.
11 C11, A9, B1, B10, A3.
```

```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
===== RESTART: C:\Users\rajuh\Documents\GitHub\Tucil-
Semester 1 : B1, B7
Semester 2 : C2
Semester 3 : A3, B10
Semester 4 : B4, A6
Semester 5 : C5, C8
Semester 6 : A9
Semester 7 : C11
>>> |
```

Figure 11 testcase

i. Keterangan

Poin	Ya	Tidak
1. Program berhasil dikompilasi	√	
2. Program berhasil <i>running</i>	√	
3. Program dapat menerima berkas input dan menuliskan output.	√	
4. Luaran sudah benar untuk semua kasus input.	√	

4. Link Source Code

[makhlukcpu/Tucil-2-Stima \(github.com\)](https://github.com/makhlukcpu/Tucil-2-Stima)

5. Referensi materi:

[Algoritma Decrease and Conquer \(itb.ac.id\)](https://itb.ac.id) Diakses pada
senin, 1 maret 2021 pukul 01.00

[Algoritma Decrease and Conquer \(itb.ac.id\)](https://itb.ac.id)

Diakses pada senin, 1 maret 2021 pukul 01.00